PlanetScale

Databases:
The Final DevOps Frontier

PlanetScale.com

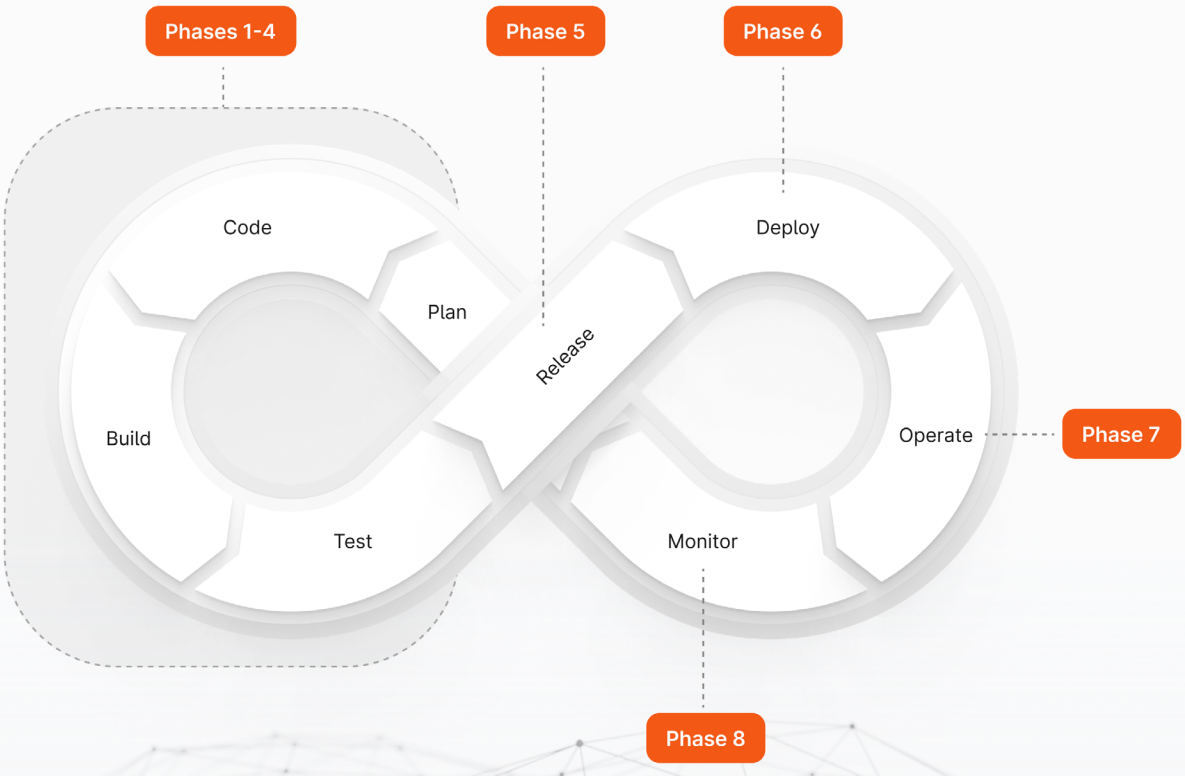


Table of Contents

Databases:
The Final DevOps Frontier

Databases:
The Final DevOps Frontier

DevOps has revolutionized the way enterprises build, test, and deploy their applications. It has transformed the way developers and operations teams work together and has led to significant improvements in speed, agility, and quality. However, there is one area of application development that has been left out of this revolution: databases. Databases are often manually managed and rarely automated, which leads to inefficiencies, security risks, data loss, downtime, and delays in application development and deployment. In this whitepaper, we will explore the reasons behind this issue and how it can be addressed by integrating databases into the DevOps process.

The DevOps Movement

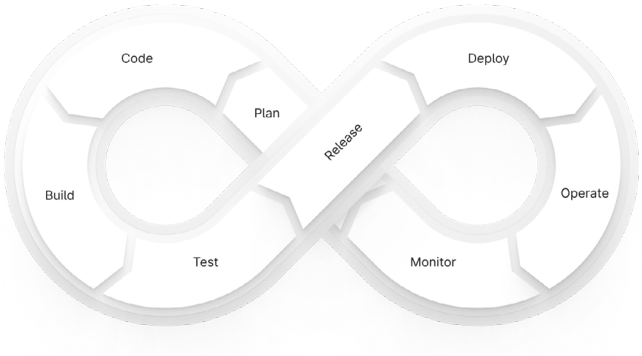
DevOps is a set of practices that combines software development and operations to build, test, and deploy applications more quickly and reliably. The movement has been revolutionary for most enterprises, as it has led to significant improvements in speed, agility, and quality. DevOps has transformed the way developers and operations teams work together, leading to more collaboration, faster feedback loops, and improved automation.

The Eight phases

There are eight distinct phases of DevOps as an operational model that enhance pipelines for a smoother development experience:

1.	**Plan**	Plan what to build and how it fits into larger business objectives.	5.	**Release**	Stage the binaries and prep for release.
2.	**Code**	Write code and build initial functionality.	6.	**Deploy**	Deploy the updated application to production.
3.	**Build**	Compile the code and create build artifacts.	7.	**Operate**	Ensure the infrastructure supports the application.
4.	**Test**	Test the artifacts to ensure integrity.	8.	**Monitor**	Gather data, feedback, and return to planning.

DevOps is not a one-size fits all model, so these eight steps act as a guide to optimize the product development process. When these phases are implemented, the flow will resemble more of an infinite loop from plan, to monitor, and back to plan as teams move through the release process.



Endless Possibilities: DevOps can create an infnite loop of release and feedback for all your code and deployment targets.

One area where DevOps has been particularly transformative is in the integration of security practices. The DevSecOps movement has emerged to address the need for security to be integrated into the DevOps process. This movement aims to ensure that security is included in all stages of the development and deployment lifecycle, from design to production. By integrating security into the DevOps process, enterprises can reduce the risk of security breaches and ensure that their applications are secure and compliant.

The Database Problem

While DevOps has been transformative for most enterprises, databases have been left out of the process. Databases are often manually managed and rarely automated. Application stakeholders often point to insufficient automation tools for database deployment (**50%**) as a major issue. Almost equally troubling are extended periods for database change approvals (**49%**) and a highly manual, error-prone deployment process (**48%**).

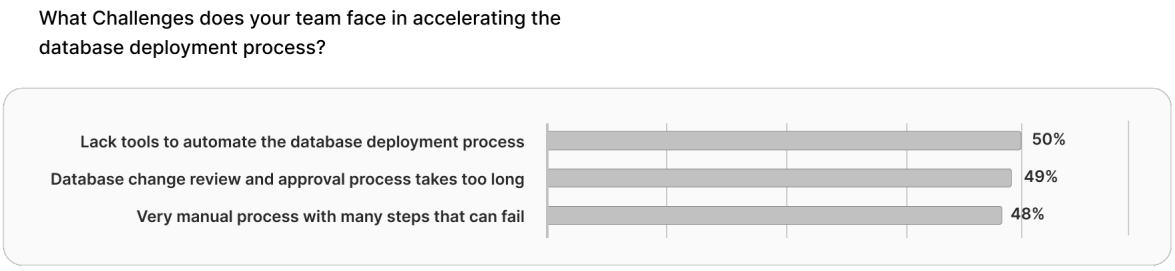
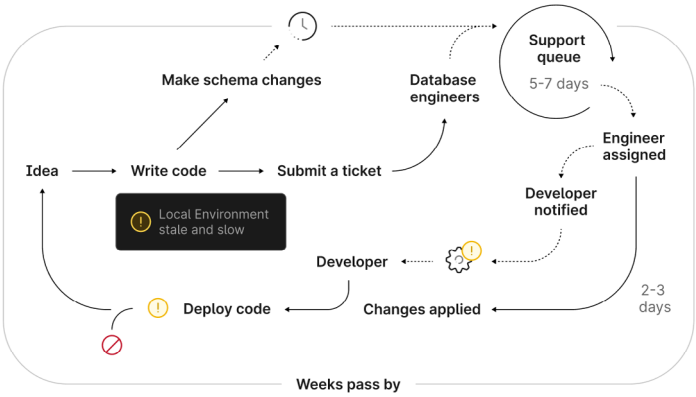


Figure 1

The problem is particularly acute when it comes to managing database schema changes. These changes are often made manually and can take a long time to propagate through the system. This can lead to delays in application development and deployment, as well as increased risk of errors and downtime.

Teams that are able to move quickly in other areas of the stack are forced to slow down significantly when interacting with the database. A meager **28%** of technology teams' time is dedicated to defining and creating new features and applications, while a staggering **72%** is consumed by managing infrastructure and accomplishing administrative tasks. This interaction involves reading documentation, opening tickets for manual DBA review, waiting for DBAs to manually access database servers to run schema changes.

- **57%** of application deployments require a schema change meaning the majority of deployments do not meet the traditional requirements of devops.
- **84%** of application stakeholders had serious production issues due to database change errors.
- **88%** report taking more than an hour to resolve these schema change issues.



[1] 2019 STATE OF DATABASE DEPLOYMENTS IN APPLICATION DELIVERY

The Solution: Integrating Databases into the DevOps cycle

This involves automating database management processes, including database schema changes, and incorporating them into the application development and deployment pipelines. By doing so, enterprises can achieve the same benefits that DevOps has delivered for application development and deployment, including faster feedback loops, improved collaboration, and greater automation.

There are several benefits to integrating databases into the DevOps process. First, it can lead to significant improvements in speed, agility, and quality. By automating database management processes, enterprises can reduce the time and effort required to manage databases, leading to faster application development and deployment. Additionally, by incorporating databases into the application development and deployment pipelines, enterprises can achieve greater automation and collaboration, leading to fewer errors and greater efficiency.

Consider a near-final build for a company using Jenkins for Continuous Integration/Continuous Deployment (CI/CD) and PlanetScale as a cloud-native MySQL database. PlanetScale's Deploy Request feature streamlines team communication regarding schema changes and allows quick, painless synchronization of database changes, preventing potential production breaks. Once thoroughly tested, Jenkins deploys these changes, using agents on production servers to fetch updates, replace binaries, apply schema changes, and finalize the working build. All of this is done in an automated fashion while streamlining team collaboration to prevent potential catastrophic errors and improve efficiency.

Another important benefit of integrating databases into DevOps is improved security. By automating security processes and incorporating them into the DevOps pipeline, enterprises can reduce the risk of security breaches and ensure that their applications are secure and compliant through a process of audited and logged change management. This is particularly important given the growing number of security threats facing enterprises today.

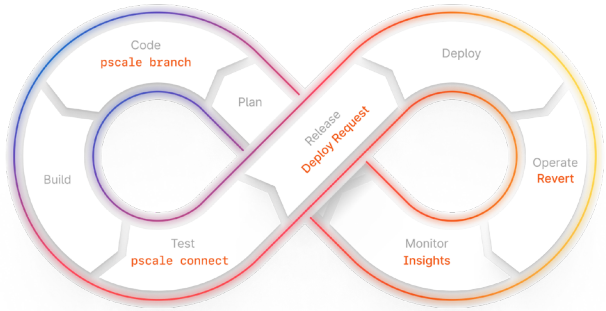
How? PlanetScale: The database for DevOps

PlanetScale is a fully managed, cloud-native, scale-out relational database management system designed specifically for applications running in the cloud. It offers a unique set of features that make it the best database for DevOps workflows. Shortly, we will explore how PlanetScale's branches and deploy requests can be seamlessly integrated into existing CI/CD pipelines to significantly increase internal developer productivity. We will also discuss how PlanetScale can roll back schema changes in the event of errors or a bad deploy.

Modern applications require a high degree of speed, agility, and scalability. Developers need to be able to rapidly write and test code, and operations teams need to be able to deploy and scale applications quickly and efficiently. In this environment, traditional databases often become a bottleneck, as they are not designed to scale horizontally and require significant manual intervention to manage.

PlanetScale Improves DevOps Processes for Databases:

This is where PlanetScale comes in. PlanetScale is the only database that is truly designed for DevOps workflows. Its cloud-native architecture, horizontal scalability, and open-source foundation make it the ideal database for modern, cloud-native applications. Along each of the eight operational steps in DevOps, PlanetScale aids in streamlining team processes while making it safer, faster, and less painful to deploy applications into production. PlanetScale's features, such as branches and deploy requests, seamless integration into existing workflows, and rollback capabilities, are designed to streamline database development and management processes and eliminate the need for manual operations.



Phases 1 through 4: The Plan, Code, Build, and Test Phases

In these phases, organizations gather to understand requirements, begin building, and then test what's been built.

Introducing methods to automate processes that would otherwise introduce tedium, risk, and take time from development teams is critical in these phases. PlanetScale plays a unique role in helping to automate the database processes via database schema branching, automating branching via the PlanetScale CLI, and other features that streamline the database change process.

Branching

PlanetScale's branches feature allows developers to work on multiple versions of the database schema in parallel while keeping them separate from each other.

> "We wanted PlanetScale to bring to MyFitnessPal what Kubernetes brought to application delivery and deployment."
>
> - Chris Karper, Engineering Director

With PlanetScale's database branches, engineers can deploy database schema like they do code, deliver it live, and PlanetScale handles the transition seamlessly. This allows developers to experiment with new ideas and features without affecting the main database schema. Branches can be merged into the main schema when they are ready, making it easy to keep the database schema up-to-date with the latest changes.

Command Line Interface (CLI) and Public API

The PlanetScale CLI or public API can enable teams to automatically create a branch whenever a new sprint begins. With this layer of automation, developers can begin making changes to the schema in an expedited and safer fashion without worrying about impacting the production environment.

There's an added security benefit to using the PlanetScale CLI as the pscale connect command can be utilized in place of sharing connection strings. This reduces the likelihood of a connection string being obtained by a bad actor and the security implications associated with this.

Phase 5: Release

Deploy Requests

PlanetScale's deploy requests feature allows developers to request a deployment of their changes to the database schema. This request is sent to the operations team, who can review the changes and approve or reject the deployment. Once the deployment is approved, the changes are automatically deployed to the database. All database deployments are done in a fully online manner. Meaning no locking, downtime, or outages.

Phase 6: Deploy

Revert

One of the most important features of PlanetScale is its ability to roll back schema changes in the event of errors or a bad deployment. This ensures that the database schema remains consistent and reliable, even in the event of errors or issues. If a deploy causes errors it can be automatically rolled back without data loss. This is a huge optimization over the traditional method of rolling back your database which involves restoring from backup, hours of downtime, and potential data loss.

Phase 7: Operate

Backups

Being able to restore your data to a snapshot in time is critical to minimize risk and data loss. This functionality is built into PlanetScale with daily backups of database branches automatically configured and included free of charge. Teams can schedule additional backups as needed. Because database branches are restored, which are isolated instances of MySQL, all of the data is restored with no data loss.

Linear and horizontal scale

> “The biggest thing for us is the **unlimited runway for growth and volume** we have thanks to PlanetScale’s ability to scale horizontally.”
>
> - Michael Ndubuisi, Engineer, Segmentation team

PlanetScale is built on top of Vitess, an open-source project that enables horizontal scale for MySQL databases. With PlanetScale, every database you spin up, whether it's free or paid, gets Vitess under the hood. This makes it easier for teams to operationalize the sharding and query routing process and ultimately scale up while increasing application performance and resiliency.

Query cache

PlanetScale Boost is a groundbreaking technology that enables you to significantly improve the performance of queries at the click of a button and without having to build tedious cache invalidation logic or maintain caching infrastructure. Useful for many of the same use cases as an application cache backed by Redis or memcached, but native to the PlanetScale platform.

Read-only regions

PlanetScale Portals gives teams the option to create read-only regions to more effectively serve queries to users in that area. Traditionally this would require operations teams to set up additional data centers linked by VPN tunnels or private ISP networks to securely synchronize data, but this is all handled by PlanetScale without such complexity.

Phase 8: Monitor

PlanetScale Insights

PlanetScale Insights is a powerful tool that allows for troubleshooting, remediation, and full lifecycle management of databases. With PlanetScale Insights, you can quickly and easily identify and troubleshoot issues with your database, from minor performance issues to critical outages. The tool provides real-time monitoring and alerts, which help to identify potential problems before they become major issues.

PlanetScale Connect

PlanetScale Connect is a feature provided to our databases that allows you to extract data from the database and safely load it into remote destinations for analytics or ETL purposes. Using Connect with our supported destinations can enable you to further process the data in any way your organization may need. This can help provide detail as to how users are using your application based on the data that's written to your database and assist in driving decisions in the next planning cycle.

CI and GitHub Actions

PlanetScale plugging into popular tools such as Jenkins, Spinnaker, and GitHub Actions is easy. PlanetScale can be seamlessly integrated into existing CI/CD pipelines enabling developers to use their existing tools and workflows to manage the database schema. Because of this, PlanetScale's workflow aligns well with the git-like model developers are familiar with. This makes it easier for teams to get caught up to speed and begin testing, building, and deploying to release new software.

Conclusion

In conclusion, it is clear that traditional manual operations for database management are inefficient, risky, and time-consuming. While many organizations have adopted DevOps methodologies to streamline their software development and deployment processes, they have not yet extended these methodologies to their databases. This has led to a siloed approach that is not conducive to agility or scalability. PlanetScale is the only database that is designed for DevOps workflows and can help organizations increase internal developer productivity. By implementing database DevOps processes and using PlanetScale, organizations can achieve a higher degree of agility and scalability, reduce the risk of errors and downtime, and respond quickly to changing market conditions and customer needs.